SECA: A STEPWISE ALGORITHM FOR CONSTRUCTION OF NEURAL NETWORKS ENSEMBLES

P. M. GRANITTO, H. D. NAVONE, P. F. VERDES and H. A. CECCATTO

Instituto de Física Rosario (CONICET-UNR)
Blvd. 27 de Febrero 210 Bis, 2000 Rosario, República Argentina

ABSTRACT: Ensembles of artificial neural networks (ANN) have been used in the last years as classification/regression machines, showing improved generalization capabilities that outperform those of single networks. However, it has been recognized that for aggregation to be effective the individual networks must be as accurate and diverse as possible. An important problem is, then, how to tune the aggregate members in order to have an optimal compromise between these two conflicting conditions. Recently, we proposed a new method for constructing ANN ensembles — termed here Stepwise Ensemble Construction Algorithm (SECA)— which leads to overtrained aggregate members with an adequate balance between accuracy and diversity. We present here a more extensive evaluation of SECA and discuss a potential problem with this algorithm: the unfrequent but damaging selection through its heuristic of particularly bad ensemble members. We introduce a modified version of SECA that can cope with this problem by allowing individual weighing of aggregate members. The original algorithm and its weighed modification are favorably tested against other methods, producing an improvement in performance on the standard statistical databases used as benchmarks.

KEYWORDS: Machine Learning, Ensemble Methods, Neural Networks.

E-MAIL: granitto@ifir.edu.ar
navone@ifir.edu.ar
verdes@ifir.edu.ar
ceccatto@ifir.edu.ar

1. INTRODUCTION

Recently, ensemble techniques have been used to improve the generalization capabilities of artificial neural networks (ANNs)[1]. The motivation for this procedure is based on the intuitive idea that by combining the outputs of several individual predictors one might improve on the performance of a single generic one. Good ensembles must have accurate but diverse members, which posses the problem of generating a set of ANNs with both reasonably good individual generalization capabilities and independently distributed predictions for the test points.

The diversity of ANNs comes naturally from the inherent data and training process randomness, and also from the intrinsic non-identifiability of the model. On the other hand, there is a trade-off between the ensemble diversity and the generalization capabilities of the individual networks. Some attempts[2,3] to achieve a good compromise between these properties include elaborations of *bagging*[4] and *boosting*[5] techniques.

In a previous work[11] we proposed a simple way of generating an ANN ensemble with members that have a good compromise between accuracy and diversity. The method (here called SECA, for Stepwise Ensemble Construction Algorithm) essentially amounts to the sequential aggregation of individual predictors where, unlike in standard aggregation techniques that combine individually optimized ANNs[6], the learning process of a new member is validated by the *overall* aggregate prediction performance. That is, the early-stopping method is applied by monitoring the generalization capabilities of the previous-stage aggregate predictor *plus* the network being currently trained (see Section 3). In this way we retain the simplicity of independent network training and only the validation process becomes slightly more involved, leading in general to some controlled overtraining ("late-stopping") of the individual networks. In this work we present a more extensive test of the proposed algorithm in the regression setting by comparing it against the well-known *bagging* technique. For this comparison we use as benchmarks the Ozone, Boston Housing and Friedman#1 statistical databases.

We also show that SECA, as any other hill-climbing optimization heuristic, may suffer of poor selections made during the optimization process. This leads to the aggregation of unusually bad ensemble members that largely deteriorate the overall ensemble performance. We propose here an improvement of the algorithm —basically consisting in weighing each network in the ensemble depending on its individual performance— which can overcome this problem. The modified method, called W-SECA, is tested on the same databases used for evaluation of the original SECA.

The organization of this work is the following: In Section 2 we discuss the so-called bias/variance dilemma, which provides the theoretical setting for ensemble averaging. In Section 3 we briefly recapitulate the original SECA[11] and give some insights to understand how this method works. In Section 4 we show empirical evidence of its effectiveness by applying it to the Ozone, Boston Housing and Friedman#1 databases. Then, in Section 5 we present the weighed version of the algorithm and discuss empirical tests using the databases mentioned above. Finally, in Section 6 we draw some conclusions.

2. THE BIAS/VARIANCE DILEMMA

Consider a set of N noisy data pairs $D=\{(t_i,\mathbf{x}_i), i=1,N\}$ generated according to $t = f(\mathbf{x}) + \varepsilon(\mathbf{x})$, where t is the observed target value, $f(\mathbf{x})$ is the true regression and $\varepsilon(\mathbf{x})$ is random noise with zero mean. If we estimate f using an ANN trained on D and obtain a model f_D, then the generalization error on a test point $(t,\mathbf{x})$ averaged over all possible realizations of the data set D and noise ε can be decomposed as:

$$\underbrace{E\left[\left(t-f_D(\mathbf{x})\right)^2 \middle| D,\varepsilon\right]}_{Error} = \underbrace{E\left[\varepsilon^2 \middle| D,\varepsilon\right]}_{\sigma_\varepsilon^2} + \underbrace{\left(E\left[f_D(\mathbf{x}) \middle| D,\varepsilon\right] - f(\mathbf{x})\right)^2}_{Bias^2} + \underbrace{E\left[\left(f_D(\mathbf{x}_i) - E\left[f_D(\mathbf{x}_i) \middle| D,\varepsilon\right]\right)^2 \middle| D,\varepsilon\right]}_{Variance}$$

The first term on the RHS (σ_ε^2) is simply the noise variance; the second and third terms are, respectively, the squared bias and variance of the estimation method. The theoretical framework for ensemble averaging is based on this bias/variance decomposition[7]. From the point of view of a single estimator, we can interpret this equation by saying that a good model $f_D(\mathbf{x})$ should be not biased, and have as little variance as possible between different realizations of D and ε. It is in general believed that the first condition is reasonably well met by ANN; however, as stated in the introduction, the second one is in general not satisfied since, even for a particular data set D, different training experiments will settle to distinct local minima of the error surface.

Aggregation is a way to take advantage of the variability of ANN mentioned above. Rewriting the error decomposition as:

$$E\left[\left(t - E\left[f_D(\mathbf{x}) \middle| D,\varepsilon\right]\right)^2 \middle| D,\varepsilon\right] = Bias^2 + \sigma_\varepsilon^2 = Error - Variance,$$

we can reinterpret this equation as follows: using the average $\Phi \equiv E[f_D|D,\varepsilon]$ as estimator, the generalization error can be reduced if we are able to produce fairly accurate models f_D (small *Error*) while, at the same time, allowing them to produce the most diverse predictions at every point (large *Variance*). Of course, these are two competing conditions, but finding a good compromise between accuracy and diversity seems particularly feasible for largely unstable methods like ANN. Several ways to generate an ensemble of models with these characteristics have been discussed in the literature[2,6,8]. In the next section we propose a new method and give some arguments that suggest why it should be effective; these ideas are later supported by empirical evidence on synthetic and real world data.

3. TUNING DIVERSITY

As suggested in the previous section, in order to improve the generalization capabilities of the aggregate predictor one must generate accurate but diverse individual networks. This can be accomplished by SECA[11]:

Step 1: Generate a training set T_1 by a bootstrap re-sample[9] from dataset D and a validation set V_1 by collecting all instances in D that are not included in T_1. Produce a model f_1 by training a network on T_1 until a minimum $e_{f_1}(V_1)$ of the generalization error on V_1 is reached.

Step 2: Generate new training and validation sets T_2 and V_2 respectively, using the procedure described in Step 1. Produce a model f_2 training a network until the generalization error on V_2 of the *aggregate* predictor $\Phi_2 = ½ (f_1+f_2)$ reaches a minimum $e_{\Phi_2}(V_2)$. In this step the parameters of model f_1 are kept constant and the model f_2 is trained with the usual (quadratic) cost function on T_2.

Step 3: Iterate the process until an optimal number N_A of models is produced. This optimal number can be estimated by keeping an external validation set or simply from the behavior of $e_{\Phi_n}(V_n)$ as a function of *n.*

In the algorithm above described the individual networks are trained with a late-stopping method based on the current *ensemble* generalization performance. The method seems to reduce the ensemble generalization error without paying much attention to whether this improvement is related to enhancing the members' diversity or not. We can see that it actually finds diverse models to reduce the aggregate error as follows[11]. Let's assume that after *n* iterations we have an aggregate predictor Φ_n, which produces an average error $e_{\Phi_n}(V_n)$ on the validation set V_n. When training model f_{n+1}, the average validation error on V_{n+1} of the aggregate predictor Φ_{n+1} will be

$$e_{\Phi_{n+1}}(V_{n+1}) = (n+1)^{-2}\,\mathrm{E}\left[\left(t - f_{n+1}(\mathbf{x}) + n\left(t - \Phi_n(\mathbf{x})\right)\right)^2 \middle| (t,\mathbf{x}) \in V_{n+1}\right]$$

$$= (n+1)^{-2}\left\{ e_{f_{n+1}}(V_{n+1}) + n^2 e_{\Phi_n}(V_{n+1}) + 2n\mathrm{E}\left[\left(t - f_{n+1}(\mathbf{x})\right)\left(t - \Phi_n(\mathbf{x})\right) \middle| (t,\mathbf{x}) \in V_{n+1}\right]\right\}.$$

In general we will have $e_{\Phi_{n+1}}(V_{n+1}) < e_{\Phi_n}(V_{n+1})$ (otherwise enlarging Φ_n would be useless) and we expect $e_{\Phi_{n+1}}(V_{n+1}) < e_{f_{n+1}}(V_{n+1})$ due to overtraining of model f_{n+1} (see next section). Then

$$\mathrm{E}\left[\left(t - f_{n+1}(\mathbf{x})\right)\left(t - \Phi_n(\mathbf{x})\right) \middle| (t,\mathbf{x}) \in V_{n+1}\right] < e_{\Phi_{n+1}}(V_{n+1}),$$

which is only possible if f_{n+1} is at least partially anticorrelated to the aggregate Φ_n.

This analysis shows that at every stage SECA is looking for a new diverse model anticorrelated with the current ensemble. In the next section we will show how this heuristic works on real and synthetic data.

4. SECA EVALUATION ON BENCHMARK DATABASES

We used three regression problems to evaluate the algorithm described in the previous section: the synthetic Friedman#1 and the real-world Ozone and Boston Housing data sets. We compared our method against the Bagging optimization strategy, which applies early stopping to each network individually. In spite of its simplicity, this methodology gives excellent results[2,4]. In order to compare the different methods' performances we used the same training process of individual networks for all of them, changing only the stopping point selection criterion. That is, any difference in performance can only be due to the construction algorithm. We set $N_A = 20$ after checking on preliminary evaluations that there was no improvement in using bigger ensembles.

The results quoted below for the Friedman#1 database correspond to an average over 50

independent runs of the whole procedure, without discarding any anomalous case. On the two real-world datasets we performed 100 runs (also without discarding anomalous cases) because the smaller test sets allow larger sample fluctuations. Notice that the indicated standard deviations only characterize the dispersion in performances due to different realizations of training and test sets; they have no direct relevance in comparing the average performances for different methods since in each run all methods use the same data.

Noise & Length	Single	Bagging	SECA	W-Bagging	W-SECA
Free, 200	1.98 ± 1.46	0.71 ± 0.47	0.65 ± 0.41	0.29 ± 0.26	**0.27 ± 0.20**
Low, 200	5.80 ± 1.05	3.66 ± 0.60	3.32 ± 0.54	3.59. ± 0.63	**3.24 ± 0.54**
High, 200	10.72 ± 1.72	7.72 ± 0.80	7.22 ± 0.63	7.65 ± 0.80	**7.18 ± 0.63**
Free, 100	7.39 ± 1.98	4.21 ± 0.61	3.97 ± 0.60	4.10 ± 0.63	**3.84 ± 0.59**
Low, 100	8.79 ± 1.41	6.16 ± 0.75	5.53 ± 0.62	6.08 ± 0.76	**5.49 ± 0.61**
High, 100	13.43 ± 1.87	10.86 ± 0.89	**10.27 ± 0.88**	10.83 ± 0.92	10.30 ± 0.96
Free, 50	9.88 ± 2.52	7.00 ± 0.50	6.84 ± 0.51	7.05 ± 0.58	**6.83 ± 0.69**
Low, 50	11.53 ± 2.79	9.20 ± 0.92	9.15 ± 0.95	9.21 ± 0.92	**9.06 ± 0.87**
High, 50	16.57 ± 3.49	13.42 ± 1.41	13.39 ± 1.42	13.40 ± 1.41	**13.32 ± 1.40**

Table 1: Mean-squared test errors averaged over 50 runs for Friedman#1 dataset, corresponding to four different algorithms for ensemble learning. The results for Single correspond to the average performance of a single ANN. The best result in each case is bolded. The standard deviations only characterize the performance fluctuations due to different realizations of training and test sets.

- *Friedman#1*

The Friedman#1 synthetic data set corresponds to training vectors with 10 input and 1 output variables generated according to

$$t = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon,$$

where ε is Gaussian noise and $x_1,..,x_{10}$ are uniformly distributed over the interval [0,1]. Notice that $x_6,..,x_{10}$ do not enter in the definition of t and are only included to check the prediction method's ability to ignore these inputs. In order to explore the performance of SECA in different situations we used different noise levels and training set lengths. The noise component was set to three levels: No noise (i.e. ε = 0, labeled "free"), low noise (ε with distribution $N(0,1)$), and high noise (ε with distribution $N(0,2)$). We generated 1200 sample vectors for each noise level and randomly split the data in training and test sets. The training sets were 50, 100 and 200 pattern length. The test set always contained 1000 examples. We considered ANNs with 6 hidden units, trained with the standard backpropagation rule with momentum.

The average mean-squared errors (MSE) on the test sets for a single ANN, the bagging technique and the original SECA are given in the first three columns of Table 1. In this case our algorithm always produced the best performance and also a reduction on the standard deviation of ensemble

errors (except for the case of 50 patterns in the training set). The improvement in performance is clearly more dependent on the length of the training set than on the noise level. We made a paired *t*-test to check whether SECA significantly outperforms Bagging. Following the standard procedure, we considered a binary variable that assumes a value of 1 when SECA is better than Bagging and 0 otherwise. If the average of this variable differs from 0.5 and the difference is enough to be statistically significant, we can affirm that one of the methods is better than the other (in particular, if the average is smaller than 0.5 then Bagging is better than SECA, and the opposite is valid when the average is bigger than 0.5). The results of the *t*-test are given in the first column of Table 2. SECA is better than bagging in all cases, except with high noise and 50 patterns where there is no statistically significant difference. For the other two noise levels and 50 patterns the advantage of SECA is also reduced but still significant. These results are in complete agreement with the MSE comparison in Table 1.

Noise & Length	**SECA**	**W-Bagging**	**W-SECA**
Free, 200	0.88 (5.37)	1.00 (7.07)	1.00 (7.07)
Low, 200	0.96 (6.51)	0.80 (5.37)	1.00 (7.07)
High, 200	0.94 (6.22)	0.90 (5.66)	0.92 (5.94)
Free, 100	0.88 (5.37)	0.82 (4.53)	0.94 (6.22)
Low, 100	0.88 (5.37)	0.90 (5.66)	0.84 (4.81)
High, 100	0.98 (6.79)	0.66 (2.26)	0.90 (5.66)
Free, 50	0.70 (2.83)	0.44 (0.85)	0.70 (2.83)
Low, 50	0.66 (2.26)	0.54 (0.57)	0.68 (2.55)
High, 50	0.54 (0.57)	0.64 (1.98)	0.58 (1.13)

Table 2: Average number of times that each algorithm outperforms Bagging on Friedman #1 database. In parenthesis we show the significance level associated with each value. The 99% and 95% confidence levels correspond to 2.58 and 1.96, respectively.

- *Ozone*

The Ozone data correspond to meteorological information (humidity, temperature, etc.) related to the maximum daily ozone (regression target) at a location in the Los Angeles basis. Removing missing values one is left with 330 training vectors, containing 8 inputs and 1 target output in each one. The data set can be downloaded by ftp (to ftp.stat.berkeley.edu/pub/users/breiman) from the Department of Statistics, University of California at Berkeley.

We have considered ANN architectures with 5 hidden units trained by the backpropagation rule with momentum. We performed a (random) splitting of the data in training and test sets containing, respectively, 100 and 230 patterns. From Table 3 we can see that the average MSE obtained with SECA is smaller than the corresponding error produced by Bagging. The result of the paired *t*-test is shown in Table 4. Again, SECA is better than Bagging with statistical significance.

Dataset	Single	Bagging	SECA	W-Bagging	W-SECA
Ozone	23.13 ± 3.10	19.71 ± 1.82	19.40 ± 1.83	19.69 ± 1.81	**19.38 ± 1.87**
Boston	19.81 ± 8.36	15.22 ± 6.20	**14.97 ± 6.16**	15.31 ± 6.26	15.07 ± 6.26

Table 3: Mean-squared test errors for two real-world databases. The results are averaged over 100 runs corresponding to four different algorithms for ensemble learning. The Single column corresponds to the average performance of a single ANN. The best result in each case is bolded. The standard deviations only characterize the fluctuations in performance due to different realizations of training and test sets.

Dataset	SECA	W-Bagging	W-SECA
Ozone	0.67 (3.40)	0.61 (2.20)	0.75 (5.00)
Boston	0.65 (3.00)	0.51 (0.20)	0.70 (4.00)

Table 4: Average number of times that each algorithm outperforms Bagging on the two real-world databases. We show in parenthesis the significance level associated with each value. The 99% and 95% confidence levels correspond to 2.58 and 1.96, respectively.

- *Boston Housing*

This data set consists of 506 training vectors, with 11 input variables and 1 target output. The inputs are mainly socioeconomic information from census tracts on the greater Boston area and the output is the median housing price in the tract. It can be downloaded from the UCI Machine Learning Repository (ftp to ics.uci.edu/pub/machine-learning-databases).

We considered 200 training examples and 106 data points for the test set. Networks with 5 hidden units were trained by the backpropagation rule with momentum. Tables 3 and 4 show the results for this case. From the MSE in Table 3 we see that SECA outperforms Bagging, which is confirmed by the results of the *t*-test in Table 4.

5. WEIGHING SECA

SECA is a stepwise optimization technique that can be classified as a hill-climbing search heuristic. A known problem with these heuristics is that during the optimization process they cannot review the choices made in the past. Figure 1 shows a typical example of this phenomenon, for a given realization of the Friedman#1 dataset. Full squares represent the evolution of training and test errors during the construction of the ensemble using SECA. In this example, the fourth added network clearly deteriorates the ensemble performance, and this effect cannot be compensated by the addition of more networks. Obviously, it also influences the selection of the following ensemble members.

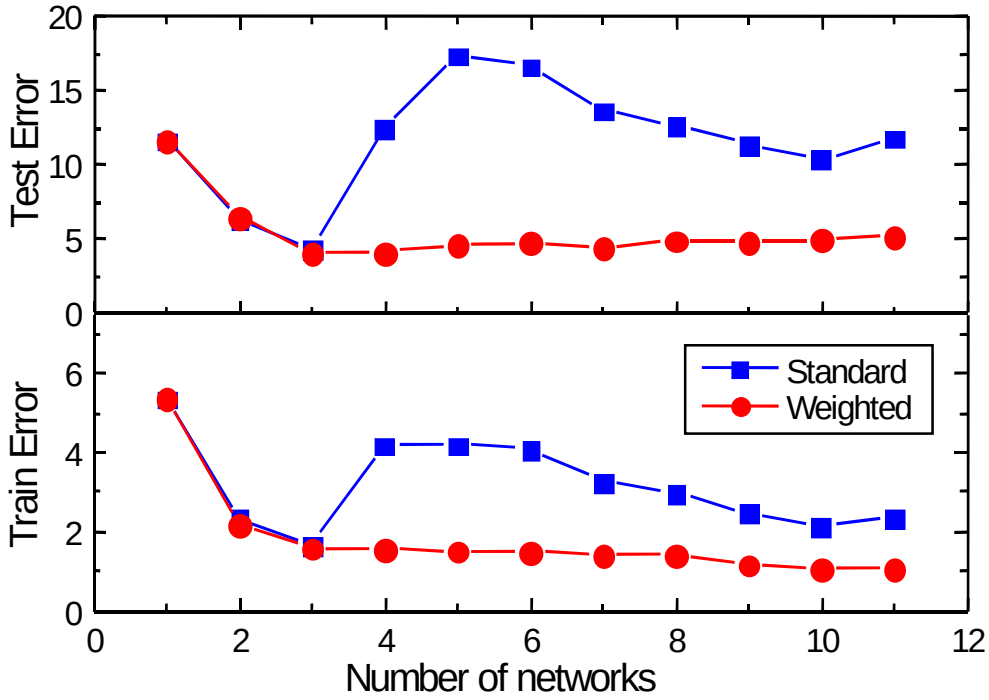

Figure 1: Evolution of train and test errors during the ensemble construction in arbitrary units.

In a previous work[10] we explored a possible way to cope with this problem, using a slightly different SECA algorithm, which only accepted networks that improved the ensemble performance. Unfortunately, new results showed that the algorithm also produced overfitting, and was unable to clearly outperform Bagging on noisy datasets. A possible intermediate solution is weighing the ensemble members, instead of rejecting them if they do not improve the overall ensemble performance. This allows us to reduce the influence of bad choices made in the past by simply giving smaller weights to troublesome networks. Then, we propose to modify the algorithm so that the output of the ensemble at the *n*-th stage becomes

$$\Phi_n(x) = \sum_{i=1}^{n} w_i f_i(x).$$

Following boosting ideas[5], we weigh each ensemble member according to its individual performance:

$$w_i = \frac{1/e_i}{\sum_j 1/e_j},$$

where e_i corresponds to the MSE of the *i*-th member over the whole training dataset. Figure 1 also shows the results of weighing: full circles correspond to the selection made by W-SECA for the case discussed above. The problematic fourth network is given a small weight, and is practically ignored by the ensemble.

To evaluate the algorithm just described we used the same three regression problems considered in the previous section. We also applied the weighing scheme to the Bagging algorithm, which allows us to check whether the possible improvement in performance is due to a better member selection process or is just a consequence of the effective elimination of some ensemble members (by giving

them small weights). Furthermore, to have a direct comparisons among all methods, we used the same overall setting considered in Section 4.

- *Friedman#1*

In the last two columns of Table 1 we can see the corresponding MSE on the test set for weighed Bagging (W-Bagging) and W-SECA. This last method produced the best performance in 8 out of 9 cases, and it was outperformed only by SECA for 100 patterns in the training set and high noise. On the other hand, W-Bagging outperformed Bagging in 7 out of 9 cases. Notice, however, that W-Bagging outperformed the standard SECA only in the simplest case, when using 200 patterns without noise. The results for the paired *t*-test against Bagging are given in the last two columns of Table 2. W-SECA is better than Bagging in all cases, and only for high noise and 50 patterns the improvement is not statistically significant. W-Bagging is better than Bagging in 7 cases. We also performed a paired *t*-test against SECA. The results are show in Table 5. The results are the same as in the MSE case, showing that W-Bagging only outperforms SECA in the simplest case.

Noise & Length	**Bagging**	**W-Bagging**	**W-SECA**
Free, 200	0.12 (5.37)	1.00 (7.07)	1.00 (7.07)
Low, 200	0.04 (6.51)	0.16 (4.81)	0.78 (3.96)
High, 200	0.06 (6.22)	0.08 (5.94)	0.62 (1.70)
Free, 100	0.12 (5.37)	0.26 (3.39)	0.70 (2.83)
Low, 100	0.12 (5.37)	0.18 (4.53)	0.64 (1.98)
High, 100	0.02 (6.79)	0.08 (5.94)	0.40 (1.41)
Free, 50	0.30 (2.83)	0.28 (3.11)	0.64 (1.98)
Low, 50	0.34 (2.26)	0.40 (1.41)	0.60 (1.41)
High, 50	0.46 (0.57)	0.52 (0.28)	0.64 (1.98)

Table 5: Average number of times that each algorithm outperforms SECA on Friedman #1 database. In parenthesis we show the significance level associated with each value. The 99% and 95% confidence levels correspond to 2.58 and 1.96, respectively.

- *Ozone*

For this dataset both weighed methods show improvements over Bagging, as can be see from Table 3. This is confirmed by the results of the paired *t*-test shown in Table 4, where both methods are better than Bagging with statistical significance. As in the Friedman#1 case, SECA is still better than W-Bagging. In Table 6 we show the results of a paired *t*-test against SECA. In agreement with the MSE results, W-SECA outperforms SECA with a confidence level bigger than 90%. SECA remains better than W-Bagging, with statistical significance.

- *Boston Housing*

The last two columns of Table 3 show the results for this case. Both weighed methods are unable to outperform the standard methods. Weighed SECA remains better than both versions of Bagging.

The paired *t*-test against Bagging, shown in Table 4, confirms these results. Both SECA algorithms outperform Bagging with statistical significance, and there is no difference between the weighed and the standard version of Bagging. In Table 6 we can see the results of the paired *t*-test against SECA. In agreement with Table 3, SECA remains better than both versions of Bagging. W-SECA shows a slight improvement over standard SECA, but the difference is not statistically significant.

Dataset	**Bagging**	**W-Bagging**	**W-SECA**
Ozone	0.33 (3.40)	0.36 (2.80)	0.59 (1.80)
Boston	0.35 (3.00)	0.37(2.60)	0.58 (1.60)

Table 6: Average number of times that each algorithm outperforms SECA on the two real-world databases. In parenthesis we show the significance level associated with each value. The 95% and 90% confidence levels correspond to 1.96 and 1.64, respectively

7. CONCLUSIONS

We performed a thorough evaluation of a simple method for balancing diversity and accuracy of ANN ensemble members. This method, that we termed SECA, seeks at every stage for a new member that is at least partially anticorrelated with the previous-stage ensemble estimator. This is achieved by applying a late-stopping method in the training process of individual networks, leading to a controlled level of overtraining of the ensemble members. The algorithm retains the simplicity of independent network training and, moreover, it largely reduces the computational burden compared to other algorithms like NeuralBAG[2] or the method proposed in [6] (which require saving the intermediate networks during training, since the selection of stopping points for the ensemble members is performed only at the end of all the training processes). Our method is a stepwise construction of the ensemble, where each network is selected at a time and only its parameters have to be saved. We showed, by comparison with Bagging, that this strategy is effective, as exemplified by the results on three standard statistical benchmarks, the Ozone, Boston Housing and Friedman#1 datasets.

We also discussed a known problem with stepwise selection procedures, and proposed a modification of the algorithm to overcome it. This modification, which we called W-SECA, consists in weighing individual members of the ensemble depending on their individual performance. We showed that it improves the results obtained with SECA in practically all cases. Moreover, the weighed version of Bagging also improved the standard version. However, only in one case out of 11 this technique outperformed SECA, and it was never better than W-SECA. This suggests that the combination of anticorrelation and weighing is the key issue of our algorithm.

REFERENCES

[1] A. J. C. Sharkey, Ed., "Combining Artificial Neural Nets", (Springer-Verlag, London, 1999)

[2] J. Carney and P. Cunningham, "Tuning diversity in bagged ensembles", *International Journal of Neural Systems* **10**, 267-280 (2000)

[3] H. Drucker, R. Schapire and P. Simard, "Improving performance in neural networks using a boosting algorithm", in S. J. Hanson, J. D. Cowen and C. L. Giles, eds., *Advances in Neural Information Processing Systems* **5**, 42-49 (Morgan Kaufman, 1993)

[4] L. Breiman, "Bagging predictors", *Machine Learning* **24**, 123-140 (1996)

[5] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", in *Proceedings of the Second European Conference on Computational Learning Theory, 23-37* (Springer Verlag, 1995)

[6] U. Naftaly, N. Intrator and D. Horn, "Optimal ensemble averaging of neural networks", *Network: Comput. Neural Syst.* **8**, 283-296 (1997)

[7] S. Geman, E. Bienenstock and R. Doursat, "Neural Networks and the Bias/Variance Dilemma", *Neural Computation* **4**, 1-58 (1992)

[8] D. Opitz and J. Shavlik, "Generating accurate and diverse members of a neural network ensemble", in D. Touretzky, M. Mozer and M. Hasselmo, eds., *Advances in Neural Information Processing Systems* **8**, 535-541 (MIT Press, 1996)

[9] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap* (Chapman and Hall, London, 1993)

[10] H. Navone, P. Granitto, P. Verdes and H. Ceccatto, "A Learning Algorithm for Neural Network Ensembles", *Revista Iberoamericana de Inteligencia Artificial* **3**, 70-74 (2001)

[11] P. Granitto, H. Navone, P. Verdes and H. Ceccatto, "Late-Stopping Method for Optimal Aggregation of Neural Networks", *International Journal of Neural Systems*, in Press